Task/Subtask IV02.1
CDRL Sequence A014–009
22 February 1995

SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS) PROGRAM

Technical Papers: Making Use of Your Defined Processes to Support Project Planning and Product Quality

Contract No. F19628–93–C–0129
Task IV02 – Megaprogramming Transition Support

Prepared for:

Electronic Systems Center
Air Force Materiel Command, USAF
Hanscom AFB, MA 01731–2116

Prepared by:

Loral Federal Systems
700 North Frederick Avenue
Gaithersburg, MD 20879

Task/Subtask IV02.1
CDRL Sequence A014–009
22 February 1995

SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS) PROGRAM

Technical Papers: Making Use of Your Defined Processes to Support Project Planning and Product Quality

Contract No. F19628–93–C–0129

Task IV02 – Megaprogramming Transition Support

Accesion For		
NTIS CRA&I		☒
DTIC TAB		☐
Unannounced		☐
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and / or Special	
A-1		

Prepared for:

Electronic Systems Center
Air Force Materiel Command, USAF
Hanscom AFB, MA 01731–2116

Prepared by:

Loral Federal Systems
700 North Frederick Avenue
Gaithersburg, MD 20879

REPORT DOCUMENTION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*

2. REPORT DATE
2/22/95

3. REPORT TYPE AND DATES COVERED
Informal Technical Report

4. TITLE AND SUBTITLE
Making Use of Your Defined Processes to Support Project Planning and Product Quality

5. FUNDING NUMBERS
F19628-93-C-0129

6. AUTHOR(S)
William H. Ett, Loral
Adam Linehan, ccPE
Jim Terrel, ccPE
Herb Krasner, Krasner Consulting

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Loral Federal systems
700 North Frederick Avenue
Gaithersburg, MD 20879

8. PERFORMING ORGANIZATION REPORT NUMBER
A014-009

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Electronic Systems Center/ENS
Air Force Materiel Command, USAF
5 Eglin Street, Building 1704
Hanscom Air Force Base, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
N/A

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Cleared for Public Release, Distribution is Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The STARS program was instituted to devlop technology to support the megaprogramming of software systems, or systems in which software is a part. Developing software using the megaprogramming approach involves following a defined process to develop software, using the concepts of architecture-based and component reuse. STARS is currently in its technology demonstration phase, where the three STARS prime contractors are each paired with a military service team to use STARS megaprogramming concepts to develop and field a software system.

Experiences by all three STARS contractors, as well as experience on all three STARS demonstration projects, have shown that defining enactable (or executable) processes is a time-consuming activity. Further, organizations are not taking full advantage of this investment from the standpoints of project and product-quality planning.

The purpose of this paper is to describe how process definitions can be leveraged to support software project planning and project execution through process-driven software development.

14. SUBJECT TERMS
Process Defintion, Project Planning, Product Quality

15. NUMBER OF PAGES
16

16. PRICE CODE
N/A

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

Preface

This document was developed by the Loral Federal Systems - Gaithersburg, located at 700 North Frederick Avenue, Gaithersburg, MD 20879. Questions or comments should be directed to Mr. William H. Ett at 301-240-6337 (Internet: ettb@lfs.loral.com).

This document is approved for release under Distribution "A" of the Scientific and Technical Information Program Classification Scheme (DoD Directive 5230.24).

The contents of this document constitutes technical information developed for internal Government use. The Government does not guarantee the accuracy of the contents and does not sponsor the release to third parties whether engaged in performance of a Government contract or subcontract or otherwise. The Government further disallows any liability for damages incurred as the result of the dissemination of this information.

Making Use of Your Defined Processes to Support Project Planning and Product Quality

by

William H. Ett, Loral Federal Systems, STARS Program
Jim Terrel, Cedar Creek Process Engineering, STARS Program
Adam Linehan, Cedar Creek Process Engineering, STARS Program
Herb Krasner, Krasner Consulting

Abstract

The Software Technology for Adaptable, Reliable Systems (STARS) program was instituted to develop technology to support the "megaprogramming" of software systems, or systems in which software is a part. Developing software using the "megaprogramming" approach involves following a defined process to develop software, using the concepts of architecture-based and component reuse. The STARS program is currently in its technology demonstration phase, where the three STARS prime contractors are each paired with a military service team to use STARS "megaprogramming" concepts to develop and field a software system.

Experiences by all three STARS contractors, as well as experiences on all three STARS demonstration projects, have shown that defining enactable (or executable) processes is a time-consuming activity. Further, organizations are not taking full advantage of this investment from the standpoints of project and product-quality planning. Process maturity assessment and process definition activities are far too often separated from project management, when, in reality, their results should be an integral part of project planning and project management activities. Processes define activities that describe work tasks, as well as verification, validation, and assessment tasks that examine a project's performance against its compliance with defined product-quality characteristics. Processes also define activities that specify the entry criteria for initiating a process, as well as completion criteria for leaving a process. Those same activities should be mirrored in our project plans to ensure that quality is built into our planning process, where activities can also be used as the basis for estimating schedule and resource requirements.

The purpose of this paper is to describe how process definitions can be leveraged to support software project planning and project execution through process-driven software development. We shall provide examples of how a state-of-the-art process management system, such as the STARS-sponsored PEAKS[1], can support the definition of processes that can be leveraged to support the above mentioned activities.

Introduction

This paper will be organized into three sections:

- Section 1 will describe characteristics that a "leveragable" process definition should possess.
- Section 2 will describe how the defined process for a project may be leveraged to support project planning, as well as address product quality concerns.

[1]PEAKS (Process Engineering and Analysis Kernel System) is a product of Cedar Creek Process Engineering of Austin, Texas. Cedar Creek Process Engineering is a member of the Loral STARS team.

- Section 3 will describe how the defined process helps project personnel satisfy the quality objectives assigned to their work products.
- Section 4 will present conclusions and describe our future plans.

Section 1 - Characteristics of the Leveragable Process Definition

Process definitions identify the work that must be performed to produce a specific set of work results, as well as how those work products will be verified and validated. Process definitions also identify the criteria required to initiate a process and those criteria required to complete it. When committed to paper, the process definition becomes part of the organization's knowledge base on how business activities addressed by the process should be performed. Once a process is defined and used by the practitioners within an organization, results from its use can be analyzed and it can be systematically improved.

One of the most critical aspects of defining processes is determining if we have defined a "good" process, with respect to existing process assurance standards, such as the SEI's CMM and ISO-9001, and whether the results from performing the process meet its stated quality objectives for product and service quality. Figure 1 illustrates an abstract process definition, based on the (E)ntry, (T)ask, (V)erification and Validation, E(X)it model. As shown in Figure 1, the process accepts required inputs and initiates its work steps after its entry criteria have been satisfied. The work steps of the process are illustrated by the "Tasks" block and the "Verify and Validate" block. After the work products and results to be produced by the process are completed, the process may terminate if its exit criteria have been satisfied. Note that the "Perform Tasks" block describes the work that must be performed to achieve the results of the process. It is not the role of a process definition to define how work tasks are to be accomplished. Also note that the "Verify and Validate Work Results" block describes how work results will be verified and validated. Every task in which work is performed must identify: 1) the agents necessary to perform the work, 2) the resources required to support the work, 3) the methods which describe how the work will be performed and 4) the artifacts the task is expected to produce, along with a description of these artifacts that describe its form and content.

Figure 1 also illustrates a few other key points. Processes may be instrumented to log selected events for historical analysis and personal process improvement, to report status and events to management and team members, and to collect measurements on both process performance and product quality. Recording how much effort a process requires, as well as how much calendar time it requires is useful for supporting both activity scheduling and effort estimation. To ensure that the process we define meets established government and commercial standards, we can assess our process against process assurance criteria, such as the SEI's CMM and ISO-9001. Once this assessment is performed, pointers from the work and verification/validation tasks should be established for those process assurance criteria to support process assurance audits. Further, where quality criteria have been established for selected process work products, pointers to appropriate product assessment criteria and checklists should be recorded and maintained. Finally, we must remember that the process was defined for a purpose, and the most vital aspect of process assurance is to ensure that the process that was defined, satisfies the requirements it was intended to address; thus pointers to the requirements for a

process should also be maintained. The navigation block shown in Figure 1 describes the rules for addressing problems found while performing verification and validation tasks. Those rules identify where to branch in the activity network to address the rework requirements caused by not satisfying specified verification and validation criteria.

Table 1 provides a summary of the characteristics a process component must possess to effectively support project planning and performance activities [Ett-93].

The description of each process should include a representation of the flow of work tasks as well as the artifacts the process must employ and produce. Using PEAKS, a task flow is represented as a constrained activity network, which illustrates the flow of tasks and their precedence constraints. An understanding of the task flow for a set of processes aids in the coordination of work between project participants.

Each process description should also contain a representation of the artifact flow required by the process, so that the artifact derivation chain is understood by developers. It provides an alternate view of the process that supports the validation of the planned work flow. Process engineers can use the artifact flow representation to ensure the planned work flow employs and produces all of the artifacts specified in the artifact flow. Once defined, the artifact flow can be integrated with the tasks required to employ and prepare the specified artifacts. For more information on the use of defined process components to prepare project processes, please refer to the paper entitled "Building Quality into Process Definitions [Ett-95]." This paper describes how the process for a project can be assembled from tailoring existing and defining new process components. We shall ask the reader to accept that process components can be used to compose larger process components, and ultimately to compose the process to support a software project.

Section 2 - Leveraging Defined Processes

The thesis of this paper is that defined processes should be the starting place for supporting the planning and estimating of a software system. The project plan that results from using the defined project process becomes the vehicle for ensuring that the software project is conducted on a process-guided basis, from the standpoint of both monitoring and controlling the project, and ensuring product quality. Further, when events occur that cause the project to replan, the process is a tool that can be leveraged to understand how to recover from those project events. Figure 2 illustrates the project planning and performance activities that can be leveraged from a defined process and a process-driven project plan. In this section, we shall provide an overview of how the project plan and the process definition from which it was derived can be leveraged by the activities identified in Figure 2.

2.1 Leveraging Defined Processes to Plan, Re-Plan and Estimate Software Projects

Planning the Software Project from a Defined Project Process

After the project process has been defined, it can be used to derive a project plan. This project plan is represented as a network of activities derived from the process, with effort and schedule information added. This scheduled activity network can be analyzed for realism with respect to project schedule and resource limitations. Using a process

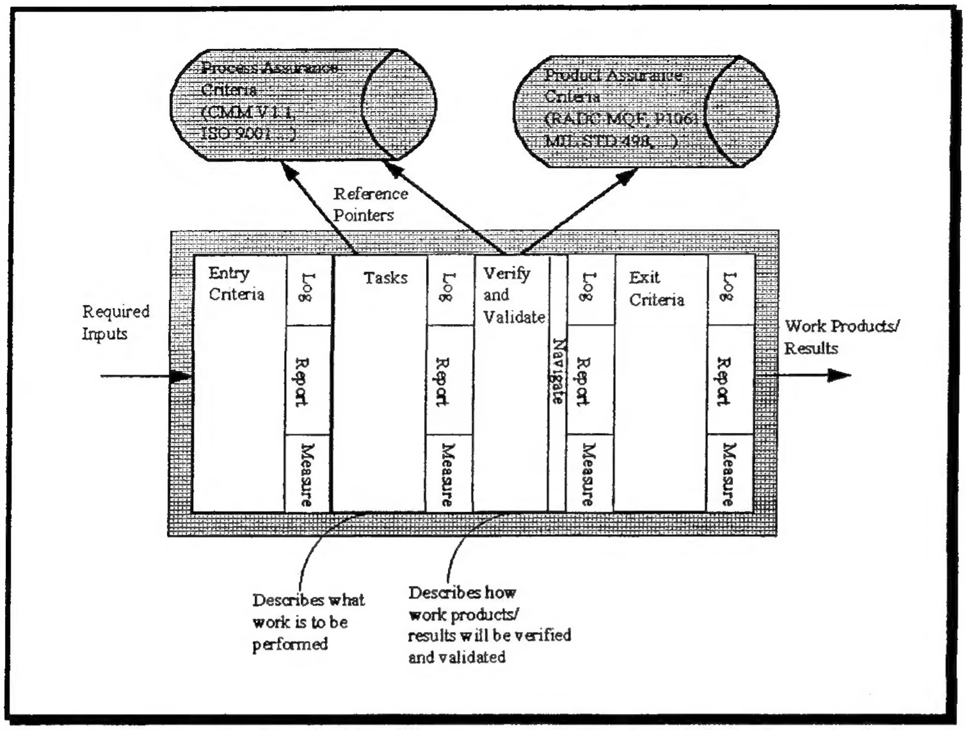

Figure 1: The (E)ntry (T)ask (V)erification and Validation E(X)it Characteristics of a Process.

Work tasks	Every process component must contain a network representation of all necessary project tasks and their constraints.
Verification and validation tasks	Every process component must contain the verification and validation tasks necessary to support the evaluation of work results produced by the process component. It also must contain pointers to the quality criteria to be used to support verification and validation tasks.
Effort estimate	Every task within a process component must identify the effort required to support it.
Resource identification	Every task within a process component must contain a pointer to the personnel and infrastructure resources necessary to support it.
Artifact identification	Every artifact to be consumed or produced by the process component must be identified, and further, effort to produce artifacts of a similar class should be recorded.
Measurement tasks	Tasks may be included in process components to identify the collection of measurements or the computation of metrics.
Status reporting tasks	Tasks may be included in process components to identify when and what status data should be reported.
Logging tasks	Tasks may be included in process components to identify when and what data should be recorded to support historical process analysis.
Navigation rules	Tasks must be included in every process component to identify how to navigate within a project process, given a process component's success or failure.
Agent identification	Every task to be performed must identify the agents required to perform the task.
Resource identification	Every task to be performed must identify the resources required to support task work.
Method identification	Every task to be performed must provide a pointer to the method that will be used to support it.

Table 1 : Characteristics of the Leveragable Defined Process

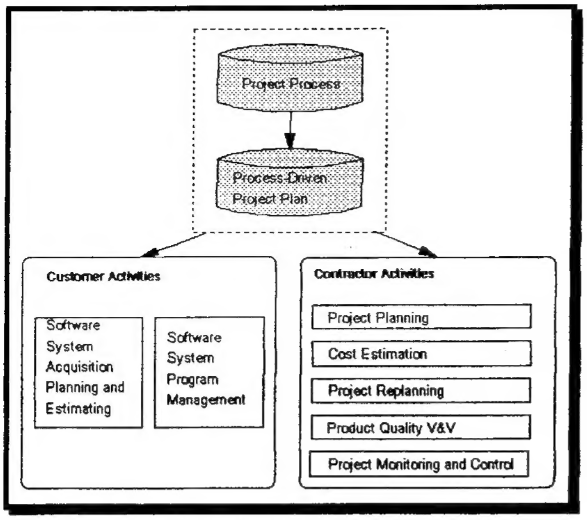

Figure 2: Activities that can be leveraged from the project process and the process-driven project plan.

management tool such as PEAKS, a project plan can easily be generated from a defined project process. As described earlier we can view a project process as an organized and integrated set of process components, which describe how a project intends to produce a set of work results. Each process component can be viewed as a generic description of how a project activity will be performed to produce a specific set of products. For example, to produce a software release for a project, the process may require the creation of release specifications, release software designs, the release software, and a release certification report. Given that the project identified that a system is to comprise three releases, a process could be instantiated for this project to create those release products for each specified release. In this way, we can generate a plan for a project from a defined process.

To illustrate how a process definition could be used to prepare a project plan, we shall show an example prepared from PEAKS [Ett-92]. Figures 3, 4, and 5 illustrate the use of the PEAKS process management tool to generate a project plan from a defined process. We refer to this as *process-driven project planning.* Figure 3 illustrates a process component for developing software releases. This process component requires three inputs, namely 1) "**REQ DEV SW REL** (request the development of a software release)," 2) a "**Validated SAS** (Software Architecture Skeleton)," and 3) a "**Validated System Specification.**" This process component consists of two work tasks, namely "**Plan SW Release** (Plan Software Release)" and "**Develop Release Software.**" After the release software is prepared it is certified ("**Certify SW Release**"), which yields the product "**Certified Software Release.**" If the certified software release passes the "**Appraise Software Release**" task, the final product of the process component is prepared, namely the "**Accepted Software Release.**" Figure 4 illustrates a simple project model for the "SCAI" project, which identifies the software system "**SCAI**" being composed of two releases, namely "**CatMaint** (Catalog Maintenance)" and "**SurvProc** (Surveillance Processing)."

Figure 5 illustrates the results from instantiating process component "Develop Software Release," where the process component activity threads are duplicated for each software release. One thread is generated for Catalog Maintenance, and one thread is generated for Surveillance Processing. Also note in Figure 5 that the "**Validated SAS**" and the "**Validated System Specification**" are *system level artifacts* produced by system level processes. The requests for developing software releases are release level artifacts and are so indicated in the plan activity network. After a project process definition is instantiated with a "*project model*" of the software products to be produced, an *unscheduled activity network* is generated.

From this discussion, we can see how a process definition could be leveraged to generate an activity network for use in supporting the project planning activities of scheduling and estimating the cost of a project plan.

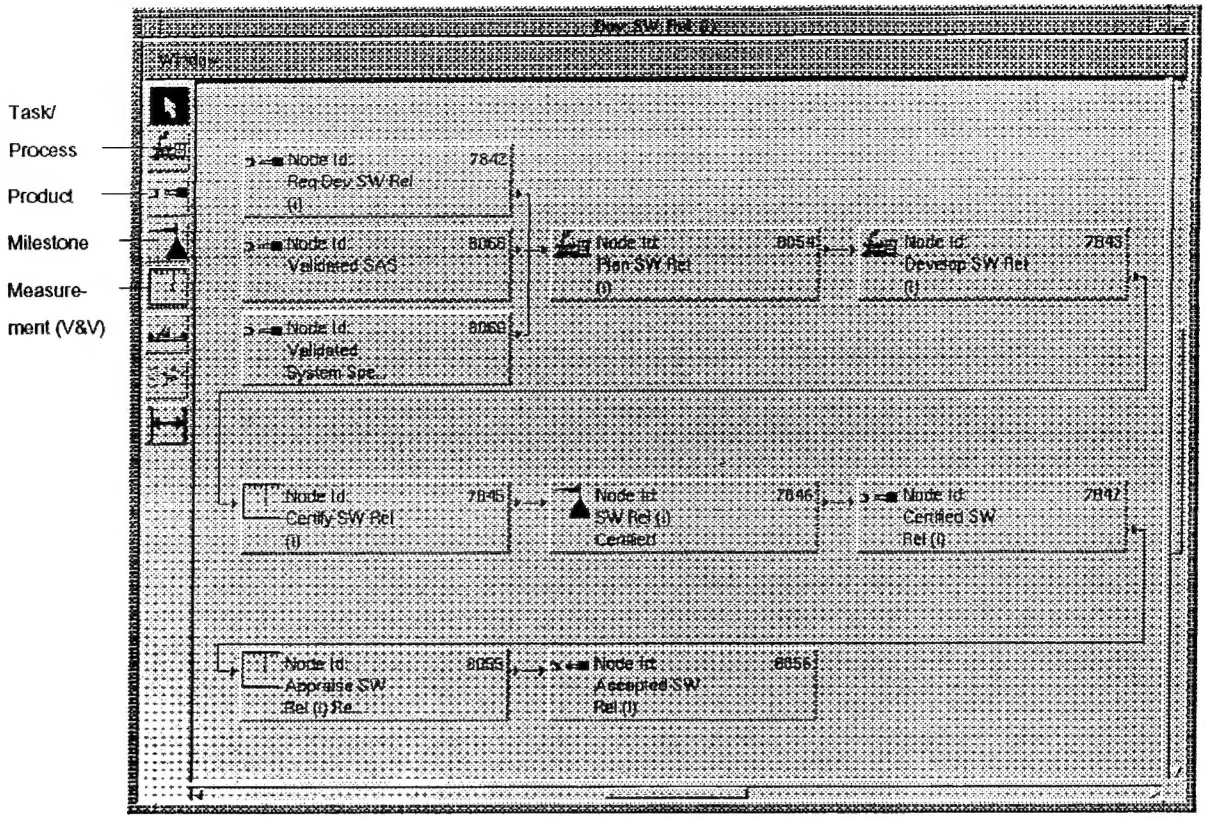

Figure 3: Process Component for "Develop Software Release"

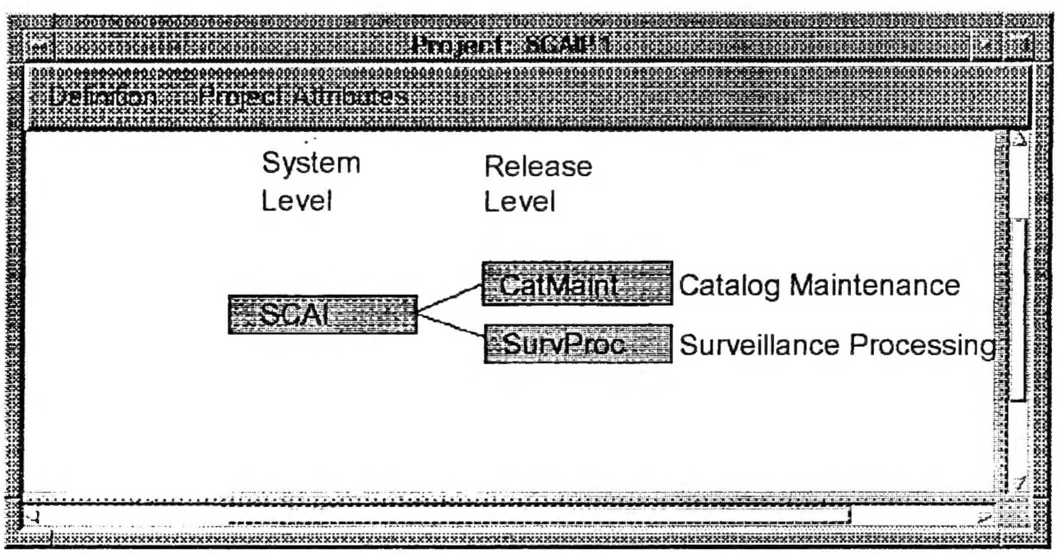

Figure 4: The project model that illustrates the products that must be created from performing the SCAI project plan."

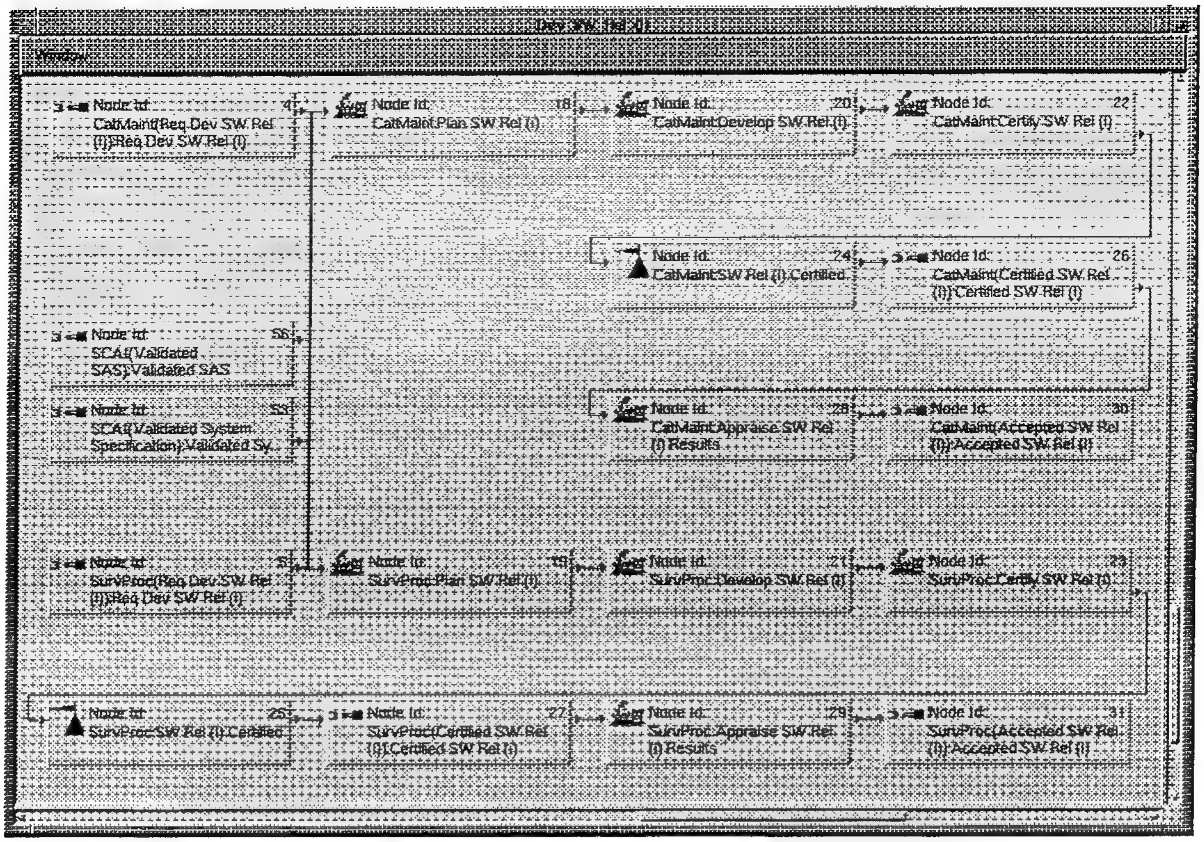

Figure 5: An instantiated process component that is part of the "SCAI Project Process."

Scheduling the Generated Project Plan

The unscheduled plan generated from a process management system such as PEAKS should be scheduled, based on the estimated: 1) effort required to produce each required software product, 2) schedule to produce each product and 3) resources required to produce and support the development and preparation of the products. PEAKS permits the project planner either to enter this data in PEAKS for plan scheduling or to export the unscheduled project plan to a project management system, such as MicroPlanner XPert[2] or CAT Compass[3]. Once the project plan is scheduled, the plan may be analyzed from both a plan and process perspective.

Preparing Data to Support Cost Estimation

Using a process management system such as PEAKS, the work tasks in a process component can be directly mapped to cost model phases and activities. Given this mapping, when effort data is applied to the work tasks of a process component, this same effort may be applied and accumulated to the associated cost model phases and activities of a selected cost model, such as COCOMO or an Activity Based Costing model.

The process definer/project planner should specify 1) the effort required to support the work tasks of a process component, 2) the resources necessary for the duration of the

[2]MicroPlanner XPert is a product of MicroPlanning, International of Mountain View, California.
[3]CAT Compass is a product of Robbins-Gioia of Alexandria, Virginia.

task, and 3) the rate of application for each resource. The role resources, e.g. personnel, hardware, software, and materials, serve as the basis for the effort estimates of a cost model and may be exported for use by the selected cost model along with the estimated effort data.

Thus, as we have discussed, the effort and schedule data applied to both the process components of a process model and the tasks of an unscheduled plan, may be leveraged to define inputs to support project cost estimation, once the project plan has been scheduled.

Validating the Project Plan for Schedule Realism

After the project plan has been scheduled, the process information associated with the plan representation may be leveraged to support project plan analysis for schedule realism. Many project plans are prepared as "success plans." By this we mean that the plans are not robust enough to tolerate unforeseen problems. Many managers place a ten to fifteen percent "management reserve" to address such problems. However, by using the process-generated project plan as a mechanism to add robustness to the project plan, the plan can be made more realistic by examining the process and identifying areas where problems might be expected due to the unprecedented nature of the system being developed, unfamiliarity with the application domain, or the introduction of new technologies and techniques to support the development of a proposed system.

Supporting project plan analysis begins with process definition. Using a process management tool such as PEAKS, process definers and project planners may instrument the verification and validation work tasks of a process component to define the evaluation characteristics that must be examined for a given product or set of products [Terr-92, Kras-92]. These evaluation characteristics permit project personnel to determine if the products they have created will pass verification and validation work tasks. An example of instrumenting a validation task with evaluation criteria is shown in Figures 6, 7, and 8. Figure 6 illustrates the selection of a measurement (or evaluation) framework to support the validation task and the data collection form selected from that framework. Figure 6 also includes a field labeled "Branch." This field is used to specify pointers to activities in the activity network in which to branch, upon a verification or validation task failure. This feature supports rework analysis. Figures 7 and 8 illustrate the selection of the measurement framework and the selection of the appropriate data collection forms. The pass/fail aspect of verification and validation activities provides project planners with the ability to "breakpoint" a project plan, much like a programmer would "breakpoint" a program. By "breakpointing" a program, the programmer can analyze the state variables of a program and interim program results. Similarly, by "breakpointing" a project plan, project planners can analyze what the effects are on a plan, given a verification or validation work task failure, and the rework required to address the failure. In this way, project plan scenarios can be prepared to indicate plan problems and the rework required to address them. The rework requirements may be factored into building in pre-planned rework cycles into the project plan, making the plan more robust. Thus, we have shown how the project plan and the process from which it was created can be leveraged to support the analysis of project plans for their schedule realism and how they could be made more robust.

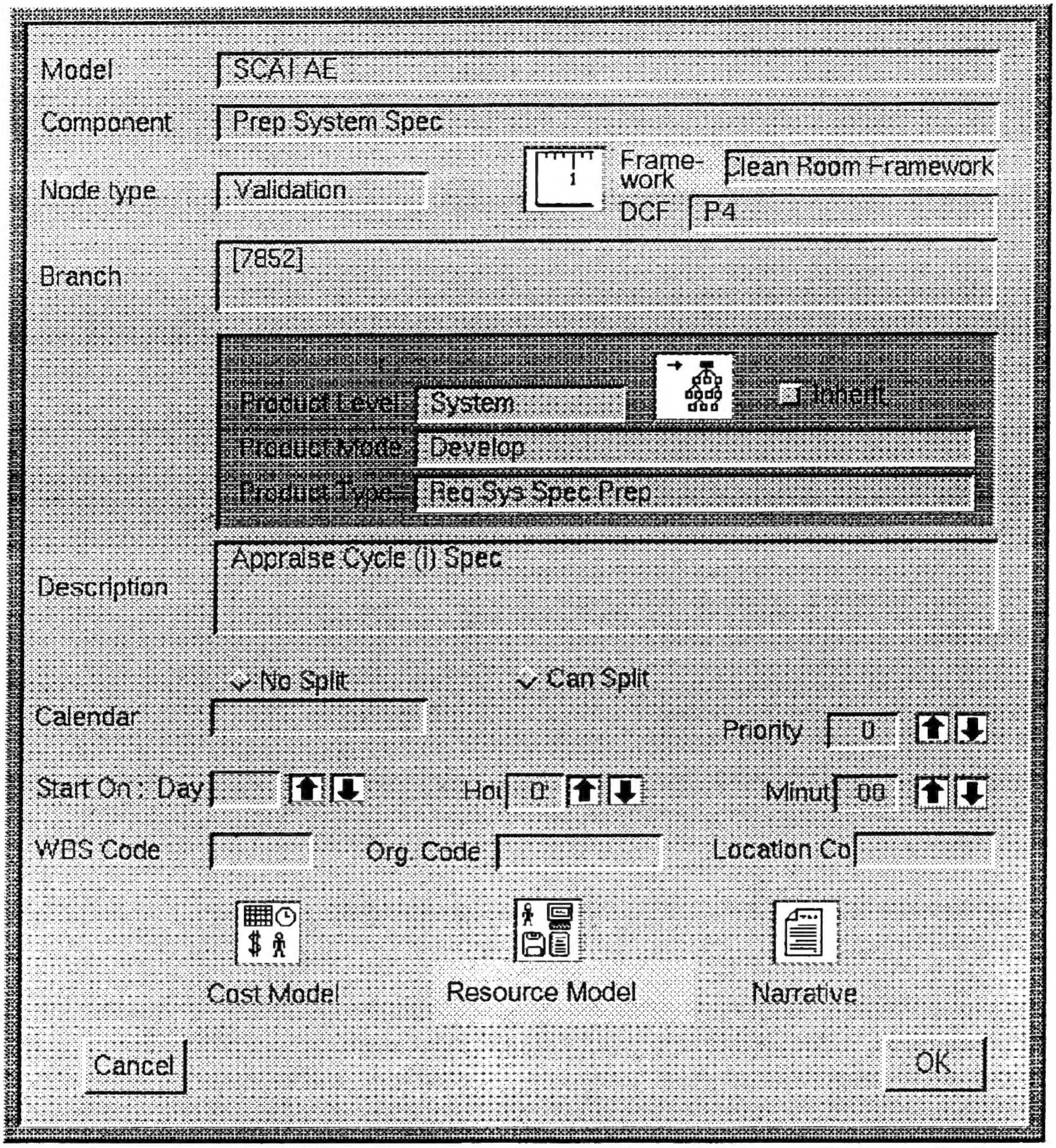

Figure 6: An example PEAKS validation work task editor. This figure illustrates 1) the measurement framework and the associated data collection form selected to support the validation task, and 2) the branching condition (or navigation rule) in the process definition, given the validation criteria are not satisfied.

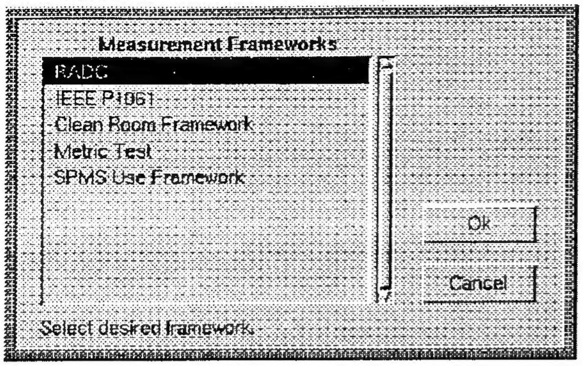

Figure 7: Illustrates the selection of the measurement (or evaluation criteria) to support the validation of a work product or result.

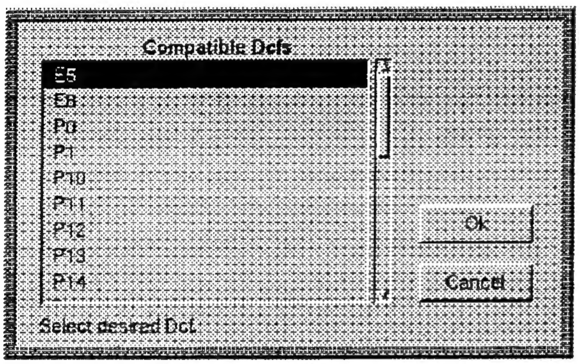

Figure 8: This figure illustrates the menu from which the process definer would select the appropriate data collection form to support the associated validation work task.

Re-planning the Software Project from a Defined Project Process

One of the requirements we identified for defining a process was the specification of navigation rules. Figure 6 illustrated where a branching condition could be specified that identified where the process would need to branch within the activity network to address a verification or validation work task failure. Using a process management tool such as PEAKS, multiple branching conditions may be specified in the verification and validation work tasks. When the project plan is declared operational and the project begins to provide status information against this plan, PEAKS will address verification and validation work task failures, based on the failure occurrence. The first occurrence of a verification or validation task failure will activate the first branching condition. The second occurrence will activate the second branching condition, etc. Thus, a process definer could decide that the first and second verification and validation failures will be handled as an internal rework cycle of a particular process. The process definer could also decide to branch to a management task for task problem review if a third verification or validation failure occurs. Because the process management system maintains a complete network of all activities and knowledge of the rework required to address verification and validation failures, the system should be capable of producing new project plans, addressing the portions of the project plan that require rework and the portions that still have not been performed. Thus, we have shown how the project plan and the process from which it was created can be leveraged to support project replanning.

Section 3 - Leveraging Process to support Process-Driven Software Development and Product Quality Assessment

Once the project plan has been accepted, and the project begins to follow the plan, the process management system can provide varying levels of support depending on the software engineering environment provided to support project work. A process mangagement system such as PEAKS, when paired with an appropriate workflow engine can:

- Provide guidance to software developers on the project's process
- Identify the products that must be produced and the requirements and quality characteristics they must satisfy
- Support product verification and validation tasks, automatically collect results from

these tasks, and use the review results to make recommendations on how to proceed

- Provide guidance on how to address a verification or validation failure, and based on the defined process, identify the tasks that must be reworked.

A process management system which includes the capabilities provided by PEAKS can be interfaced with any commercial workflow engine to automatically receive status from a specially prepared set of programs which assist project personnel in following the process defined using the process management tool. We refer to these programs as "process programs." Where process programs are instrumented to report status data, project events and employ tools to support verification and validation tasks, such as the PEAKS measurement quality facility (MQF), this data can automatically be reported to PEAKS. This permits project management to employ PEAKS as a system to support project monitoring and control, and decision support, where data about the project is automatically collected and made available for report generation and project status review. Where data is automatically reported to PEAKS, "condition watchers" may be set up to examine the PEAKS database and new transactions for unusual conditions. When the "condition watchers" identify an anomaly, they can report it to management for action in a timely manner. Examples of watchers that might be set up are to identify schedule anomalies, such as a task passing its late start window, and cost anomalies, such as identifying "earned value" problems.

Section 4 - STARS Experiences and Conclusions

One of the goals of the STARS program was to produce a process management system that supported the concepts of process-driven project planning and process-driven software development and management. Another of the project's goals was to provide an infrastructure in which to pull together activities and data used to support project planning and project management. In this way we could tie more closely the disciplines of process definition, project planning, project cost estimation, and project monitoring and control. Our work on STARS with PEAKS and its forerunner SPMS (Software Process Management System) indicates that we are heading in a positive direction to achieve these goals.

We wrote this paper to provide the reader with some examples of how the process definition prepared for a project could be leveraged to support a number of important project planning and performance activities. To date we have practical field experience in process definition and process-driven project planning. We have positioned ourselves in the STARS program, through the efforts of the Loral STARS Team and Cedar Creek Process Engineering, to test all the concepts described in this paper on future projects at the U.S. Army's Picatinny Arsenal and on the Air Force/STARS Demonstration Project (SCAI Project) in Colorado Springs, Colorado. Our plans are to do just that, as well as to transition our process management concepts and technology into business units of Loral Federal Systems.

From our work we have concluded that process-driven project planning can and does work and that it can become a driving force in the planning of projects that wish to employ the concepts of megaprograming. Further, we have concluded that "quality" must be built into our process definitions, along with the activities required to support it, so that those activities will appear in our project plans, and

thus ensure that both government and contractor personnel understand: 1) the process by which a product will be created; 2) the evaluation characteristics by which those products will be assessed; and 3) the project plan that addresses how the above will be satisfied. Our ultimate hope is that organizations will recognize that the project plan and the process definition from which it was derived can be leveraged to develop quality software within a plan that all parties understand and believe.

References

Ett-92 Ett, W., J. Terrel and H. Krasner, "Using SPMS to Support Process Modeling and Project Planning," Process Tools/Notations Report, STARS CDRL 4024-001,Task IT15.2, July 24, 1992.

Ett-93 Ett, W., "A Comparison of Simple Process Forms against Minimum Criteria for an Enactable Process," IBM Technical Report 85.0196, Gaithersburg, Maryland, February 1993.

Ett-95 Ett, W., and J. Terrel, "Building Quality into Process Definitions," Draft Technical Report, Loral Federal Systems, Gaithersburg, Maryland, February, 1994

Kras-92 Krasner, H., J. Terrel, A. Linehan, P. Arnold, and W. Ett, "Lessons Learned in Prototyping and Use of a Software Process Modeling System," Communications of the ACM, Special Issue on Modeling and Analysis, September 1992.

Terr-92 Terrel, J. and H. Krasner, "Defining, Simulating and Validating Software Process Models in SPMS," Proceedings of the National Security Industrial Association Eighth Annual National Conference on Software Quality and Produc tivity, Arlington, Virginia, March 10-12, 1992.

About the Authors

William Ett is currently working on the ARPA STARS Program, where he specializes in developing and transitioning techniques to define enactable processes, and the design and development of automated process support applications. Mr. Ett's accomplishments include the co-invention of the ProjectCatalyst front end and its generic process programming paradigm, the design of the initial LORAL STARS process support environment, and the co-development of the "STARS/SEI Process Definition Information Organizer Templates." His research interests include the design of project and process support technology to assist organizations in benefitting from process-driven software development. He may be reached at Loral Federal Systems, 700 North Frederick Avenue, Gaithersburg, MD 20879 ettb@lfs.loral.com.

Jim Terrel is the technical lead for PEAKS development and enhancement in support of the Air Force/STARS Demonstration Project. Mr. Terrel is the managing partner of Cedar Creek Process Engineering (ccPE). ccPE was formed to commercialize the Software Process Management System (SPMS) prototype developed as part of the Loral STARS contract. His research interests are in practical process management technology and its relationship to software engineering environments. He may be reached at ccPE, P.O. Box 308, Cedar Creek, TX 78612 terrelj@source.asset.com.

Adam Linehan is a senior technical staff member and partner of Cedar Creek Process Engineering. Mr. Linehan was responsible for the design and development of several PEAKS subsystems. His research interests are in practical management technology and its relationship to hypermedia interfaces. He may be reached at ccPE, P.O. Box 308, Cedar Creek, TX 78612.

Herb Krasner is President of Krasner Consulting and specializes in evaluating, planning and implementing computer software quality and process improvement programs. He is a consultant to the SEMATECH Consortium, as well as other commercial and federal clients. His research interests involve applying TQM and process management techniques to improve process and quality programs in progressive software organizations. He may be reached at Krasner Consulting, 1901 Ringtail Ridge, Austin, TX 78746, hkrasner@cs.utexas.edu.